NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

AUTONOMOUS UNDERWAY REPLENISHMENT AT SEA FOR RIVERINE OPERATIONS

by

William F. Addison II

March 2008

Thesis Advisors: Fotis Papoulias
Oleg Yakimenko

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
March 2008

3. REPORT TYPE AND DATES COVERED
Master's Thesis

4. TITLE AND SUBTITLE Autonomous Underway Replenishment at Sea for Riverine Operations

5. FUNDING NUMBERS

6. AUTHOR(S) William F. Addison II

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES)
N/A

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (maximum 200 words)

Currently the United States Navy is making a small footprint in the world's littoral regions with the help of the United States Marine Corps. In Iraq, the Marine Corps is actively conducting Riverine operations, however they are overly tasked and in need of permanent replacement by the United States Navy. In order to alleviate the Marine Corps, the Naval Expeditionary Combat Command with its Riverine Squadrons will soon take over these Riverine operational commitments in order to reestablish supremacy throughout the Riverine environment. With this in mind, the Chief of Naval Operations, Center for Naval Analyses requirements, System Engineering Analysis (SEA-11) class of 2007 developed a concept of operations (CONOPS) which the Total Ships System Engineering (TSSE) class of 2007 used to develop a prototype platform, which met all initial design requirements. In order to take full advantage of this prototype platform, every effort was taken in order to minimize the number of crew members on station at any given time. The purpose of this thesis is to demonstrate the use of the direct method, which will allow the Specialized Command and Control Craft (SCCC) to conduct a fully autonomous Underway Replenishment at Sea (UNREP) with a standard supply vessel. The direct method approach allows for a smooth path is created instead of using waypoint navigation. Additionally, this method allows for real-time updates at (1Hz).

14. SUBJECT TERMS AUV, UUV, robotics, trajectory planning, path planning, rendezvous, real-time, direct methods

15. NUMBER OF PAGES
71

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UU

THIS PAGE INTENTIONALLY LEFT BLANK

AUTONOMOUS UNDERWAY REPLENISHMENT AT SEA FOR RIVERINE OPERATIONS

William F. Addison II
Lieutenant, United States Navy
B.S., United States Naval Academy, 2001

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
March 2008

Author: William F. Addison II

Approved by: Fotis Papoulias
Thesis Advisor

Oleg Yakimenko
Thesis Advisor

Anthony J. Healey
Chairman, Department of Mechanical and Astronautical Engineering (MAE)

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Currently the United States Navy is making a small footprint in the world’s littoral regions with the help of the United States Marine Corps. In Iraq, the Marine Corps is actively conducting Riverine operations, however they are overly tasked and in need of permanent replacement by the United States Navy. In order to alleviate the Marine Corps, the Naval Expeditionary Combat Command with its Riverine Squadrons will soon take over these Riverine operational commitments in order to reestablish supremacy throughout the Riverine environment. With this in mind, the Chief of Naval Operations, Center for Naval Analyses requirements, System Engineering Analysis (SEA-11) class of 2007 developed a concept of operations (CONOPS) which the Total Ships System Engineering (TSSE) class of 2007 used to develop a prototype platform, which met all initial design requirements. In order to take full advantage of this prototype platform, every effort was taken in order to minimize the number of crew members on station at any given time. The purpose of this thesis is to demonstrate the use of the direct method, which will allow the Specialized Command and Control Craft (SCCC) to conduct a fully autonomous Underway Replenishment at Sea (UNREP) with a standard supply vessel. The direct method approach allows for a smooth path is created instead of using waypoint navigation. Additionally, this method allows for real-time updates at (1Hz).

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I. INTRODUCTION 1

A. BACKGROUND 1

B. MOTIVATION 4

1. Manning 4

2. General 5

C. SCOPE 6

D. PROBLEM FORMULATION 7

E. THESIS STRUCTURE 9

II. THE TIBERINUS CLASS 11

A. SPECIALIZED COMMAND AND CONTROL CRAFT DESCRIPTION .11

B. EQUATIONS OF MOTION 13

III. DIRECT METHOD FOR RAPID PROTOYPING 21

A. HISTORICAL BACKGROUND 21

1. General Description of Direct and Indirect Method 21

2. History 22

B. MATHEMATICAL DEVELOPMENT 25

C. HYPOTHETICAL TRAJECTORIES 30

IV. TRAJECTORY DEVELOPMENT AND VALIDATION 33

A. INDENTIFIYING CONTROL PARAMETERS & CONSTRAINTS 33

B. INVERSE DYNAMICS 33

C. OPTIMIZATION 35

V. RESULTS AND CONCLUSIONS 39

A. RESULTS 39

B. CONCLUSIONS 45

APPENDIX. MATLAB CODE 47

A. STARTMESCCC.M 47

B. TRAJECTORY.M 48

C. PLOTRESULTS.M 50

D. TEST.M 51

E. SUPPLYSHIP.M 51

F. SCCCSHIP.M 51

LIST OF REFERENCES 53

INITIAL DISTRIBUTION LIST 55

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

Figure 1. Blue Water to Brown Water Navy....................2
Figure 2. Bridge Layout.....................................4
Figure 3. Problem Outline...................................8
Figure 4. SCCC and MMC.....................................11
Figure 5. Splitting Original Interval......................24
Figure 6. Basic First Order Polynomial.....................27
Figure 7. Curved Trajectory Using Waypoints................28
Figure 8. Visual Check.....................................31
Figure 9. Random Trajectory Generation.....................32
Figure 10. General Block-Scheme.............................36
Figure 11. Multiple SCCC Scenario...........................40
Figure 12. Close-up of Initial Position of Multiple Scenario.......................................41
Figure 13. Close-up of Final Position of Multiple SCCC Scenario.......................................42
Figure 14. Controls for First SCCC..........................43
Figure 15. States for First SCCC............................44
Figure 16. Virtual Domain Parameters for First SCCC........45

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

Table 1. Ship Characteristics............................12
Table 2. Example Inputs..................................31
Table 3. Initial & Final Positions.......................37
Table 4. Penalty Weights.................................38

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like thank my thesis advisors, Professor Fotis Papoulias for his efforts in allowing me to take on my adventure, which is this thesis; and Professor Oleg Yakimenko, for whom this thesis could not have been completed without his expertise, guidance, help, time, and most of all his patience. I would not be able to claim success without their help.

I would particularly like to thank my wife, Tamika D. Addison, for supporting and encouraging me. Without her efforts at home with our wonderful children, I would not have had the time to devote to my educational endeavors.

I would additionally, like to thank my father, William F. Addison I and my sister, Samai A. Morris for supporting and their continued emotional support.

Finally, I would like to thank my mother, Pissmai Meyers, who never allowed me to ever underestimate myself and has always pushed me to succeed in my educational journey.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. BACKGROUND

When the end of the “Cold War” with the Soviet Union came about, there was a major shift in the United States Naval Doctrine. With the United States Navy’s major opponent on the high seas eliminated, so to was the threat of fighting a major naval battle on the high seas as was the threat in years past. After numerous studies and analysis of current naval operations and assets, in August 2005 the Chief of Naval Operations (CNO) announced that the United States Navy would reconstitute a Riverine capability, allowing the United States Navy to transition from a blue water navy to a force which would be capable of sustaining operations in the littoral regions of the world. The CNO’s vision called for the resurgence of the brown water Riverine Force which is called out in the CNO’s Concept of Operations (CONOPS) for the 21st Century Riverine Force. This document calls for the formation of a Naval Expeditionary Combat Command, which requires a Riverine force as one of its elements. The primary mission for this force is to conduct Phase 0 (shaping and stability) operations, to provide maritime security and to carry out additional tasks specifically related to the Global War on Terrorism. [1]

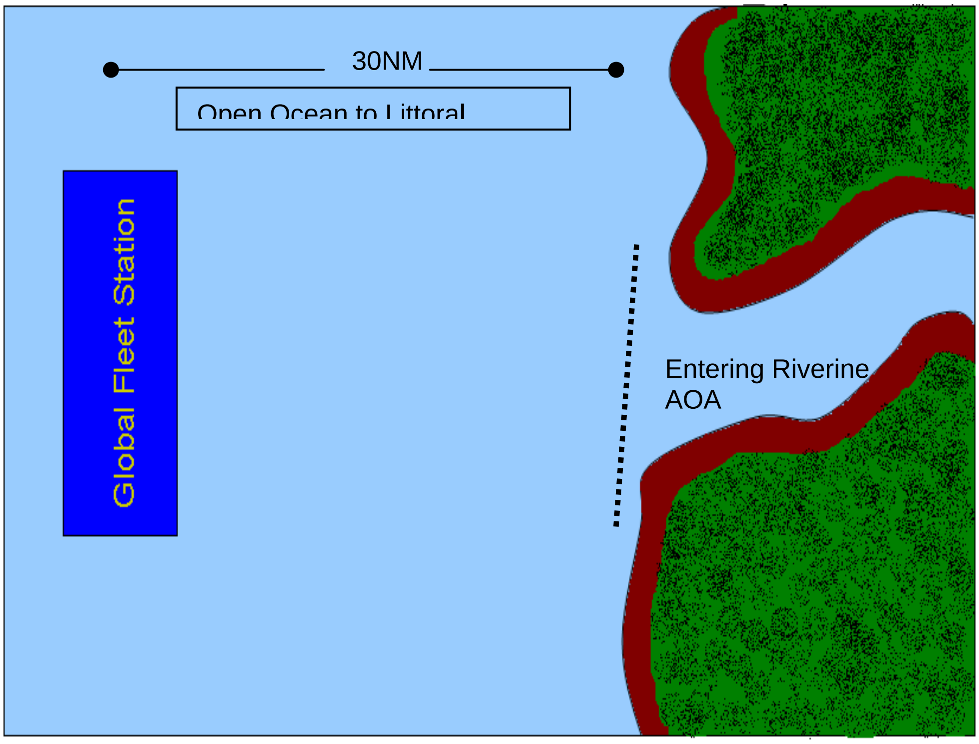

Figure 1. Blue Water to Brown Water Navy

Currently the United States Navy is making a small footprint in the world's littoral regions with the help of the United States Marine Corps. In Iraq, the Marine Corps is actively conducting Riverine operations, however they are overly tasked and in need of permanent replacement by the United States Navy. In order to alleviate the Marine Corps, the Naval Expeditionary Combat Command with its Riverine Squadrons will soon take over these Riverine operational commitments in order to reestablish supremacy throughout the Riverine environment.

Based on the Chief of Naval Operations, Center for Naval Analyses requirements, System Engineering Analysis (SEA-11) class of 2007 developed a concept of operations (CONOPS) which the Total Ships System Engineering (TSSE) class of 2007 used to develop a prototype platform, which met or exceeded all initial design requirements. The first step in the development of this new platform, was to conduct a Capability Gap Analysis of existing Naval assets both US and foreign. It was quickly determined from this analysis, that no current ships were capable of fulfilling all of the initial requirements requested and a new platform would be needed to accomplish the vision of the CNO. Based off of these results, a functional Element Decomposition of the system requirements was developed and a preliminary design was identified. The ultimate design evolved into a multi-hulled Specialized Command and Control Craft (SCCC), which would utilize three multi-mission craft (MMC) to accomplish all mission requirements.

The existing procedure for conducting Riverine operations is to first establish a land forward operating base and to then deploy Riverine assets from this land based support center to carryout various missions. While Iraq has shown a land basing system to be effective, in the future it maybe more likely that the Navy will require a sea based support structure in order to accomplish its Riverine mission. To accomplish this future scenario, the Navy could use existing assets; however, this approach limits the Navy’s future Riverine footprint due to the limited access current assets have in the majority of the rivers of the world due to these assets’ slow speeds and deep drafts. To structure the United States Riverine forces in such a way as to create maximum operational flexibility in the majority of the world’s littoral regions, this new platform must be produced.

B. MOTIVATION

1. Manning

The Riverine forces will comprise of 800 personnel divided among three squadrons. Minus the command structure for the Riverine force, each squadron allotted 225 personnel. Each squadron will consist of three SCCCs and nine MMCs. The TSSE Manning Study resulted in the need for 216 personnel to fully man these ships, with the remaining nine comprising the command structure of the squadron. The command structure will remain afloat on the Global Fleet Station (GFS) in order to manage the logistics of the SCCC. The GFS will be removed from the Riverine Area of Operation and in Blue water.

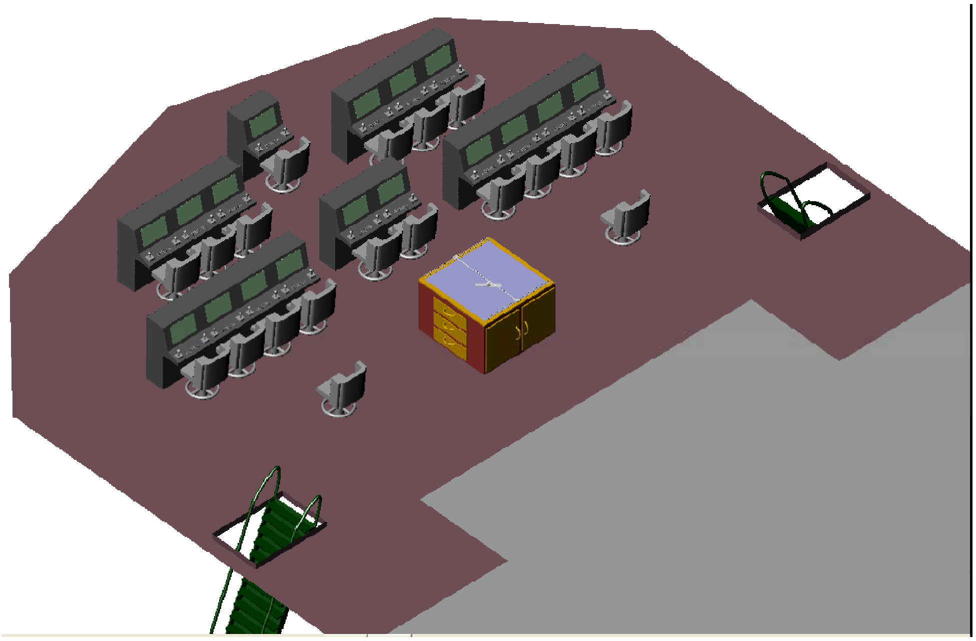

Figure 2. Bridge Layout

Every effort has been made in the design of the Tiberinus Class to minimize the number of crew members on station at any given time. With this in mind, the bridge will be the only actively manned space on the ship. As with existing naval ships, the bridge has been organized to allow the commanding officer to oversee all aspects of navigation; however, the bridge on the Tiberinus Class has also incorporated the Combat Information Center and Damage Control Central, in an effort to centrally locate all controlling stations. All bridge watch stations will be equipped with touch screen panels which will enable any watch stander to reassign their watch station to receive additional information from any other watch station and take control from their console. Figure 2 shows a purposed bridge layout. With the changes from traditional naval watch team structure, it will require a crew of 72 personnel to fully man the Tiberinus SCCC and the three associated MMCs.

2. General

Due to the reduced size and complexity of the Tiberinus Class, significant advances in automation of processes and procedures had to be achieved in order to allow its reduced crew to fully operate the ship in efforts to achieve all mission objectives. As a result, all efforts have been made to allow the ship’s crew to operate the ship with minimal personnel on station.

Currently there are no options for ships to conduct Underway Replenishment at Sea (UNREPS) operations autonomously. The ability to accomplish this would allow for increased force flexibility and operation. Additionally, if not used for fully autonomously UNREPS, this technology could be used as visual cueing for complex formation maneuvers, UNREPS, plane guard operations, or even pier dockings. This technology could be incorporated with a heads up display, which would use standard maneuvers to build a database of near-optimal trajectories calculated beforehand. These near-optimal trajectories would allow the Officer of the Deck (OOD) to not just mentally visualize the command, but this technology would allow the OOD to actually see a simulation of where he or she will end up, thus adding to the overall situational awareness.

C. SCOPE

The scope of this thesis will consist of analytically developing a path-planning process which will generate trajectories for an UNREP between a standard USNS oiler and the SCCC. The first step in achieving this objective will be to develop a hydrodynamic model of the SCCC in order to utilize the equations of motion in which it will be used to simulate the vessel movement.

The next step will be to formulate the rendezvous trajectories based off of mathematical basis. With this information, the factors which will effect the trajectories shape can be explained and constraints can be formulated to take into account both permissible trajectories and the vessel constraints. From here, this information will be used to generate the performance index of the vessel.

After the trajectories are computed, the inverse dynamics will then be used to calculate the required states of the vessel at each point upon the trajectory path. In order to minimize any violations of the optimal parameters, the values generated by the trajectory algorithm and the inverse dynamics will be used in the performance index.

D. PROBLEM FORMULATION

The problem can be summed up as follows: The supply vessel will provide a rendezvous point. From this point, as an example they will suggest a course of 090 degrees, speed 13 knots; however, as is often the case, they may need to maneuver in order to avoid a contact or to create a better UNREP situation. As the supply vessel maneuvers, they will send updates to the SCCC so that it may make real-time updates in order to achieve station at a lateral separation distance of 140 ft, while maintaining C090/S13kts.

Once this is achieved, a laser range find system will be employed to maintain the lateral separation. This is illustrated in the figure below.

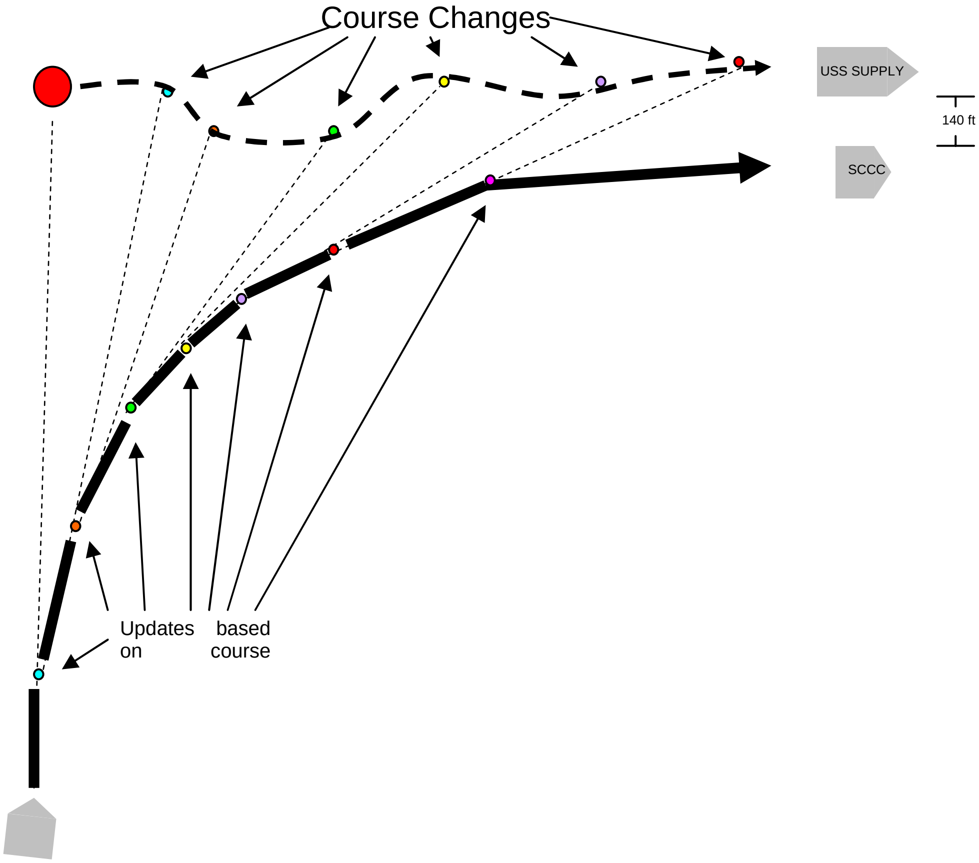

Figure 3. Problem Outline

The figure above shows a pictorial representation of the problem as described above. The proposed sequence of events is to have the supply vessel communicate with the SCCC in order to command the SCCC to proceed to the rendezvous point. The SCCC will then compute the necessary trajectory to complete the mission and reply with an acknowledgement or the SCCC will decline the command request. A denial from the SCCC would constitute a violation in one of the system constraints and a request by the SCCC would then be sent in order to allow the SCCC to reach the rendezvous point by either altering the time of arrival or by requesting a different rendezvous point all together. This update would be achieved at a rate of (1 Hz) or real-time.

E. THESIS STRUCTURE

The intent of this research is to develop a direct method control for the SCCC in order to allow for an autonomous UNREP. This will be conducted by first determining the equations of motion, followed by the development and validation of the trajectories, and finally by the vessel simulations.

Chapter II will focus on the Tiberinus Class. It will focus on the equations of motion for the SCCC and the vessel simulation development. Chapter III explains the theory and equations used for the direct method for rapid prototyping.

Chapter IV will focus on the development and validation of the trajectories for the SCCC. Chapter V will present a simulation for the UNREP between the SCCC and a supply vessel. Additionally, Chapter V will provide the thesis conclusions.

THIS PAGE INTENTIONALLY LEFT BLANK

II. THE TIBERINUS CLASS

A. SPECIALIZED COMMAND AND CONTROL CRAFT DESCRIPTION

The Tiberinus Class Ship has been designed to provide command, control and support for Shaping and Stability operations. In support of the previous mentioned operations, the SCCC will provide Maritime Security and carry out additional tasks specifically related to the Global War on Terrorism (GWOT) throughout the littoral regions of the world. By design, the Tiberinus Class will provide a sea-based maritime capability which will enable U.S. forces to have an enhanced presence in their areas of operation, will maintaining the legitimacy and sovereignty of the United States’ ally and coalition partners lands.

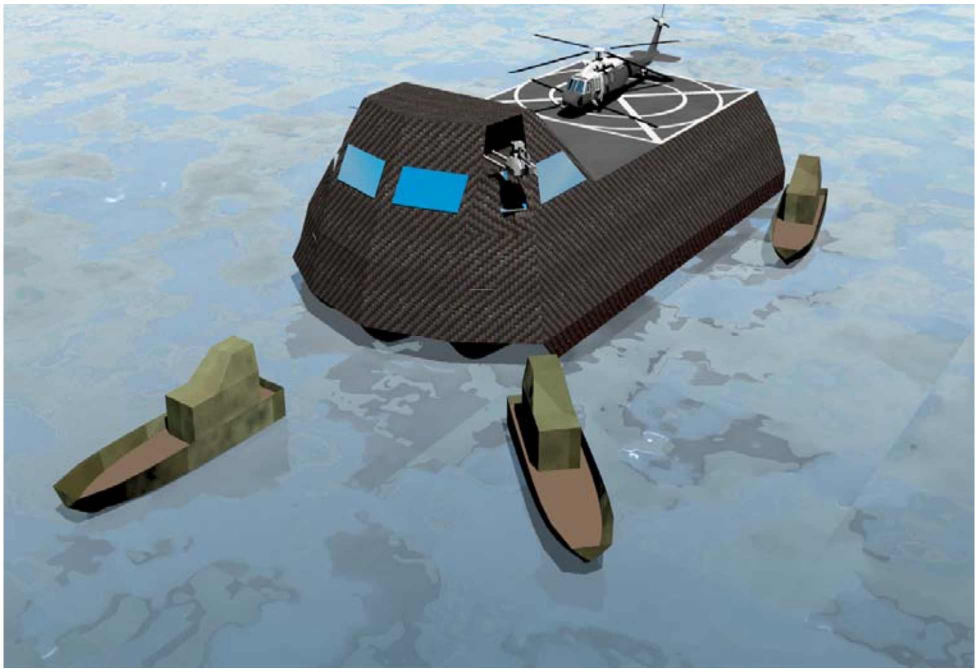

Figure 4. SCCC and MMC

This class has been designed to allow for a Riverine force to sustain a forward presence within a Riverine environment anywhere in the world for an indefinite period of time, while maintaining the capability of conducting interdiction operations, low intensity combat operations, Visit Boarding Search and Seizure (VBSS), maritime security operations, and waterborne checkpoints.

Each Tiberinus Class Ship has been fully designed to be independent of each other and will provide all of its own hotel services for its embarked personnel. Although the Tiberinus Class has primarily be designed in a supporting role, each ship has been designed to allow it to carryout the same missions as the MMCs in reference to a combat role. The characteristics of this class are as shown in the table below.

Characteristics	
Length (LOA)	135 ft
Beam	68 ft
Speed	40+ kts
Draft	6.2 ft
Range (Design)	1,500 nm
Range (Maximum)	3,750 nm
Displacement (Design)	550 LT
Aircraft	1 H-60 (landing, not housed)
Mission Craft	3 JMECs / MMCs

Table 1. Ship Characteristics

B. EQUATIONS OF MOTION

In designing the model for this vessel, the first step is to determine the equations of motion. Since the SCCC and the supply vessel will be operating in two dimensions, only equations in the horizontal plane will be considered, however, this process will be described for a three dimensional application for future iterations. This is a somewhat common equation development and will be briefly described below.

When dealing with relative motion certain assumptions must be established in regards to the motion boundaries. For the purposes of this research, it will be assumed that the vessel will act as a rigid body, which will enable for the calculation of forces and moments on the vessel. Additionally, it will be assumed that the Earth’s rotation is negligible in regards to acceleration components of the vehicle’s center of mass. This will allow for the illusion that the vessel will be moving over a stationary plane. Finally, it will be assumed that the forces acting on the vessel will have their origins in an inertial and/or a gravitational prospective. Additionally, the other primary forces action on the vessel will be hydrostatic, propulsion, thruster, and hydrodynamic forces from lift and drag.

For vehicles and vessels described in terms of three dimensional components, the velocity of these vehicles/vessels will be accounted for using six terms given as surge, sway, heave, roll, pitch, and finally yaw. For the purpose of this research, surge (u) is the vessels forward speed; sway (v) is the side slip velocity, heave (w) corresponds to a velocity component in the local Z direction, however, its global velocity components do depend on the vessels heading, pitch, and roll. These three components makeup the body fixed coordinate as shown below.

$$\dot{R}_0 = T(\alpha)[ui + vj + wk] \qquad (2\text{-}1)$$

The other three terms are represented of the angular rate of rotation of the bodies fixed frame ω as shown below.

$$\omega = [pi + qj + rk] \qquad (2\text{-}2)$$

The vector quantity of these additional three terms, have their defined meanings in terms of the vessel motion. The vessel “roll rate” is described by the p component in equation (2-2). The vessel “pitch rate” is described by the q component in equation (2-2). Finally, the “yaw rate” of the vessel is described by the r component as seen in equation (2-2). These particular components would be sensed by the onboard gyro of the SCCC. All of these components combined, account for the overall velocity of the vessel, which can be displayed as shown below or as later displayed in Chapter III.

$$x = [u\ v\ w\ p\ q\ r]^T \qquad (2\text{-}3)$$

All of the applied external loads for the body coordinate components are represented by the vector components of forces which are applied on the body of the vessel and the moments which are also applied at the center of the body fixed frame as shown below.

$$m\left\{\frac{d\mathbf{v}}{dt} + \dot{\boldsymbol{\varpi}} \times \rho_G\right\} + m(\boldsymbol{\varpi} \times \boldsymbol{\varpi} \times \rho_G + \boldsymbol{\varpi} \times \mathbf{v}) - \mathbf{f}_g = \begin{bmatrix} X_{app} \\ Y_{app} \\ Z_{app} \end{bmatrix} \qquad (2\text{-}4)$$

$$\mathbf{I}_o\dot{\omega}+\omega\times\left(\mathbf{I}_o\omega\right)+m\left\{\rho_G\times\dot{\mathbf{v}}+\rho_G\times\omega\times\mathbf{v}\right\}-\mathbf{m}_g=\begin{bmatrix}K_{app}\\M_{app}\\N_{app}\end{bmatrix} \qquad (2\text{-}5)$$

$$F(t)=[X_{app}(t)\ \ Y_{app}(t)\ \ Z_{app}(t)\ \ K_{app}(t)\ \ M_{app}(t)\ \ N_{app}(t)]^T \qquad (2\text{-}6)$$

Due to the SCCC being in its first iteration of design, the hydrodynamic coefficients were computer simulated in order to determine the necessary hydrodynamic coefficients for the equations of motion for the vessel. For the purpose of this thesis, the constants and hydrodynamic coefficients for the SCCC are not included due to the non-trivial nature of the task and the unconventional hull shape and type.

It is often difficult to assess the vessels mass moments of inertia about its Center of Gravity (CG), due to the CG change with loading and unloading of the vessel. These loading and unloading changes are usually symmetric, which is ideal when attempting to maintain the vessels proper trim and heel under normal static conditions. Typically, the mass and angular motion of the vessel is described through the mass moment of inertia matrix, which is shown below.

$$I_0=\begin{bmatrix}I_{xx} & I_{xy} & I_{xz}\\I_{yx} & I_{yy} & I_{yz}\\I_{zx} & I_{zy} & I_{zz}\end{bmatrix} \qquad (2\text{-}7)$$

The elements in the above matrix can be determined by the following equations below.

$$I_{xx}=\sum_{i=1}^{N}dm_i(y^2+z^2) \qquad (2\text{-}8)$$

$$I_{yy} = \sum_{i=1}^{N} dm_i (x^2 + z^2) \qquad (2\text{-}9)$$

$$I_{zz} = \sum_{i=1}^{N} dm_i (x^2 + y^2) \qquad (2\text{-}10)$$

$$I_{xy} = I_{yx} = -\sum_{i=1}^{N} dm_i (xy) \qquad (2\text{-}11)$$

$$I_{xz} = I_{zx} = -\sum_{i=1}^{N} dm_i (xz) \qquad (2\text{-}12)$$

$$I_{yz} = I_{zy} = -\sum_{i=1}^{N} dm_i (yz) \qquad (2\text{-}13)$$

With the angular velocity vector express as a column vector as is shown below,

$$\omega = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \qquad (2\text{-}14)$$

the angular momentum of the vessel can then be expressed as:

$$H_{\text{o}} = I_{\text{o}} w \qquad (2\text{-}15),$$

which will allow one to utilize Newton's second law in order to achieve the following equation below.

$$M_0 = \frac{d\mathbf{H}_o}{dt} + \rho_G \times \left(m \frac{d^2\mathbf{R}_G}{dt^2} \right) \qquad (2\text{-}15)$$

The right hand term in equation (2-15) is representative of the moment of the inertial reaction of the sum of the external forces acting on the vessel.

The vertical plane for the equations of motion will begin with setting all horizontal plane variables to zero, in order to allow only w, q, θ, and z to be the only variables of concern. This, with the addition of a constant speed and utilizing small angular changes will allow one to utilize a linear system approach. The reduced equations are shown below.

$$m\dot{w}_r = mU_0 q + (W - B)\cos\theta + \delta Z_f(t) \qquad (2\text{-}16)$$

$$I_{yy}\dot{q} = (Z_B B - Z_G W)\sin\theta + \delta Z_f(t) \qquad (2\text{-}17)$$

$$\dot{\theta} = q \qquad (2\text{-}18)$$

One can then manipulate the above equations to create a more useable format, which is shown below in the next set of equations that will allow the user to conduct matrix operations and finally display the state and control matrices.

$$M_v \dot{x}_v(t) = a_v x_v(t) + b_v \delta_s(t) \qquad (2\text{-}19)$$

$$m_v = \begin{bmatrix} m - Z_{\dot{w}} & -Z_{\dot{q}} & 0 \\ -M_{\dot{w}} & I_{yy} - M_{\dot{q}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (2\text{-}20)$$

$$a_v = \begin{bmatrix} Z_{\dot{w}} & \mathrm{mU}_o + Z_{\dot{q}} & 0 \\ M_{\dot{w}} & M_q & z_B B - z_G W \\ 0 & 1 & 0 \end{bmatrix} \qquad (2\text{-}21)$$

$$b_v = \begin{bmatrix} Z_{\delta_s} \\ M_{\delta_s} \\ 0 \end{bmatrix} \qquad (2\text{-}22)$$

$$\dot{x}_v(t) = A_v x_v(t) + B_v \delta_s(t) \qquad (2\text{-}23)$$

$$xv(t) = [w \;\; q \;\; \theta]^T \qquad (2\text{-}24)$$

$$Av = \begin{bmatrix} m - Z_{\dot{w}} & -Z_{\dot{q}} & 0 \\ -M_{\dot{w}} & I_{yy} - M_{\dot{q}} & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} Z_w & mU_0 - Z_q & 0 \\ M_w & M_q & z_B B - z_G W \\ 0 & 1 & 0 \end{bmatrix} \qquad (2\text{-}25)$$

$$B_v = \begin{bmatrix} m - Z_{\dot{w}} & -Z_{\dot{q}} & 0 \\ -M_{\dot{w}} & I_{yy} - M_{\dot{q}} & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} Z_w \\ M_w \\ 0 \end{bmatrix} \qquad (2\text{-}26)$$

Disregarding the motions in the vertical plane will result in the horizontal equations of motions, which are shown below.

$$m\dot{v}_r = -mr + Y_{\dot{v}_r}\dot{v}_r + y_{v_r}v_r + Y_{\dot{r}}\dot{r} + Y_r r + Y_\delta \delta_r(t) \qquad (2\text{-}27)$$

$$I_{zz}\dot{r} = N_{\dot{v}_r}\dot{v}_r + N_{v_r}v_r + N_{\dot{r}}\dot{r} + N_r r + N_\delta \delta_r(t) \qquad (2\text{-}28)$$

$$\dot{\psi} = r \qquad (2\text{-}29)$$

As with the previous section in regards to the vertical plane, the horizontal dynamic equation is

$$m_h \dot{x}_h(t) = a_h x_h(t) + b_h \delta_r(t) \qquad (2\text{-}30)$$

where

$$x_h(t) = [v \; r \; \psi]^T \qquad (2\text{-}31)$$

Again, as previously shown, this equation converted into a more functional form and the state and control matrices result as shown below.

$$\dot{x}_h(t) = A_h x_h(t) + B_h \delta_r(t) \qquad (2\text{-}32)$$

$$A_h = \begin{bmatrix} m - Y_{\dot{v}} & -Y_{\dot{r}} & 0 \\ -N_{\dot{v}} & I_{zz} - N_{\dot{r}} & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} Y_v & Y_r - mU_0 & 0 \\ N_v & N_r & 0 \\ 0 & 1 & 0 \end{bmatrix} \qquad (2\text{-}33)$$

$$B_h = \begin{bmatrix} m - Y_{\dot{v}} & -Y_{\dot{r}} & 0 \\ -N_{\dot{v}} & I_{zz} - N_{\dot{r}} & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} Y\delta_r \\ N\delta_r \\ 0 \end{bmatrix} \qquad (2\text{-}34)$$

THIS PAGE INTENTIONALLY LEFT BLANK

III. DIRECT METHOD FOR RAPID PROTOYPING

A. HISTORICAL BACKGROUND

1. General Description of Direct and Indirect Method

From many years of research, it has been determined that the most precise approach in solving an optimal control problem is by the variational approach, which is based on the Pontryagin’s minimum principle. This indirect approach requires the solution of the necessary conditions of optimality associated with the infinite dimensional optimal control problem rather than optimizing the cost of a finite dimensional discretization of the original problem directly. Using this method does require advance analytical skill in which one must generate numerical solutions of the resulting two-point boundary-value problem. The minimum principle is used to eliminate the controls in this indirect method approach, resulting in a generally nonlinear function of the state and co-state variables. This indirect approach allows for the generation of benchmark solutions, which will generally converge if only excellent initial guesses for the non-intuitive con-states are achieved. Additionally, this requires the switching structure to be guessed correctly in advance.

A direct method approach allows for rapid trajectory prototyping ability. This method uses finite dimensional discretization of the optimal control problem to a nonlinear programming problem. While the direct method approach does not allow for extremely great precision and resolution as that of the indirect method, it does allow for a more practical application as its convergence robustness is far superior.

2. History

The ideal of the direct method approach was first developed by Euler in the early 1900's, when he approached the solution of functions as finite sets of variables. This approach was achieved by representing acceptable functions in the form of infinite power series

$$y(x) = \sum_{k=0}^{\infty} a_k x^k, \text{ (3-1)}$$

or by Fourier series

$$y(x) = \frac{a_0}{2} \sum_{k=1}^{\infty} (a_k \cos kx + b_k \sin kx), \text{ (3-2)}$$

or by any series in the form of

$$y(x) = \sum_{k=1}^{\infty} a_k \varphi_k(x), \text{ (3-3)}$$

where $\varphi_k(x)$ is a given function. Thus, instead of an infinite series, the user is only considering a finite series, whose solution is simply the function of a set of unknown coefficients.

Ritz too developed a direct method, in which his method requires a field problem to be arranged, in which it will be used as an integral minimization. Thus, allowing it to be used for problems which have variational principles. Galerkin obtained approximate solutions to boundary-value problems in a simpler way, which is why it is more of a universal process. When Galerkin's method is combined with the interpolation equations of the method of finite elements, it allows one to solve both initial and boundary-value problems, which Galerkin utilized to solve parabolic and elliptic partial differential equations.

In the 1950's, aerospace engineers began to utilize the finite element method. Due to vast improvements in computing in the following year, this method became popularized for numerous numerical simulations of physical problems dealing with stress analysis, structural and solid mechanics, heat transfer, and fluid mechanics among others. Resulting conclusions found that the previous fore mentioned methods when applied, will yield approximation for the minima from above or the maxima from below, thus, enabling the user to utilize them for rapid prototyping of optimal solutions or near-optimal solutions. This allows for the ability to preset extreme trajectories and/or controls, while allowing for a calculational advantage, while providing a near-optimal solution with any varying degree of accuracy.

In the 1960's, Taranenko applied a similar method to that Ritz-Galerkin to flight dynamics involving constraints on states and controls. Continuing in his predecessors' methods, he attempted to use continuous, unequivocal and differentiable functions which automatically satisfied boundary conditions of the function

$$x_i = x_{i0} + \frac{x_{if} - x_{i0}}{\tau_f - \tau_0}(\tau - \tau_0) + \Phi_i(\tau), \text{ i=1,...4} \qquad (3\text{-}4)$$

as the reference function for the Cartesian coordinates and speed. In equation (3-4), τ is an argument, while $\Phi_i(\tau)$ is a continuous, unequivocal, and differentiable function which satisfies the boundary conditions $\Phi_i(\tau_0) = \Phi_i(\tau_f) = 0$. Taranenko further suggested the uses of the following equations or any of their linear combinations.

$$\Phi_i^1(\tau) = \sum_{k=1}^{n} A_k \sin k\pi \frac{\tau - \tau_0}{\tau_f - \tau_0} \quad (3\text{-}5)$$

$$\Phi_i^2(\tau) = \sum_{k=1}^{n} A_k (\tau - \tau_0)^k (\tau - \tau_f)^k \quad (3\text{-}6)$$

$$\Phi_i^3(\tau) = (\tau - \tau_0)^{m_1} (\tau - \tau_f)^{m_2} \quad (3\text{-}7)$$

While there are no actual limitations, one could use any convenient function for their particular task. State parameters and controls can then be resolved from the result of their inverse flight dynamics.

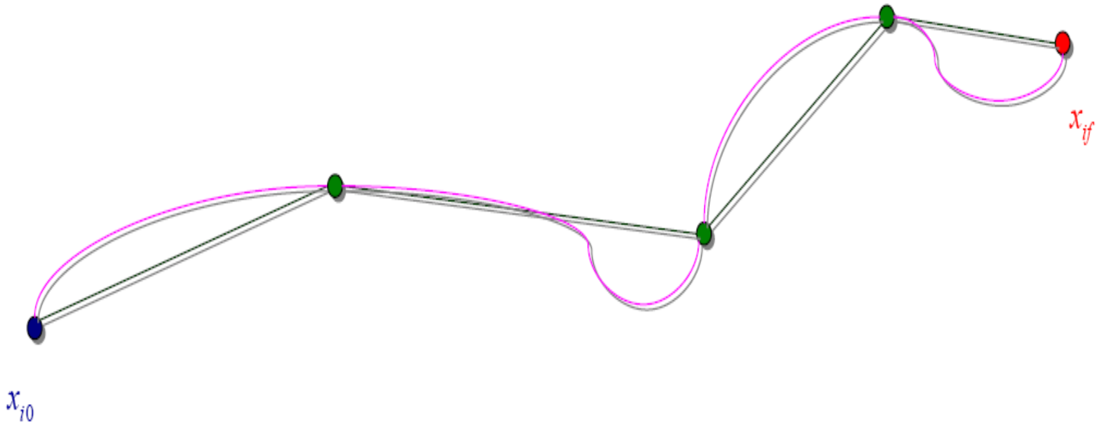

Figure 5. Splitting Original Interval

In order to provide more flexibility without increasing the order $n\,(m_1, m_2)$, Taranenko recommended the splitting of the interval $[\tau_0;\tau_f]$ into pieces, in order to use lower order polynomials in order to describe the behavior of the state variables $x_i (i = 1,2,3,4)$. The higher order $n\ (or\ m_1 \text{ and } m_2)$, the higher the number of pieces required in the piecewise case, thus the closer a near-optimal solution will be to the optimal solution.

Taranenko continued his research with the hopes of building a database of trajectories for numerous aircraft, in an effort to aid pilots in maneuvering their aircrafts, by suggesting the maneuvers optimal trajectory. Due to the lack of computing speed of the day, it was discovered that the numerous optimization parameters would not allow for onboard computation of these trajectories.

As with most discoveries, this approach was overlooked until technology could appropriately catch-up with the computing requirements. In 1997, Taranenko’s dream was realized by Yakimenko who tested these methods onboard a flying laboratory. Yakimenko developed an algorithm which computes near-optimal trajectories. These trajectories were found to be capable of satisfying all boundary conditions prior to the establishment of the trajectories, which allowed for decreased calculational time and the ability to conduct real time onboard trajectory computation.

B. MATHEMATICAL DEVELOPMENT

In order to develop the mathematics for this problem, the first step is to formulate an optimal control problem which will move the vessel from a starting point (initial point) to some final point. If one chooses or is given both an initial and final position, which will also serve as the problems boundary conditions, a first order polynomial representation of a trajectory linking both positions can be developed by using the following formulas:

$$x(\tau) = P_X(\tau) = a_0 + a_1\tau \qquad \text{(3-8)}$$

$$y(\tau) = P_y(\tau) = b_0 + b_1\tau \qquad \text{(3-9)}$$

As previously mentioned, τ is given as any argument. One can then solve for any unknown coefficient by substituting any value for τ and solving for these unknown coefficients with the following matrix equations:

$$\begin{bmatrix} 1 & 0 \\ 1 & \tau \end{bmatrix} \begin{bmatrix} a_0 \\ a_f \end{bmatrix} = \begin{bmatrix} x_0 \\ x_f \end{bmatrix} \qquad \text{(3-10)}$$

$$\begin{bmatrix} 1 & 0 \\ 1 & \tau \end{bmatrix} \begin{bmatrix} b_0 \\ b_f \end{bmatrix} = \begin{bmatrix} y_0 \\ y_f \end{bmatrix} \qquad \text{(3-11)}$$

$$y = y_0 + \left[\frac{y_f - y_0}{x_f - x_0} \right] \left(x - x_0 \right) \qquad \text{(3-12)}$$

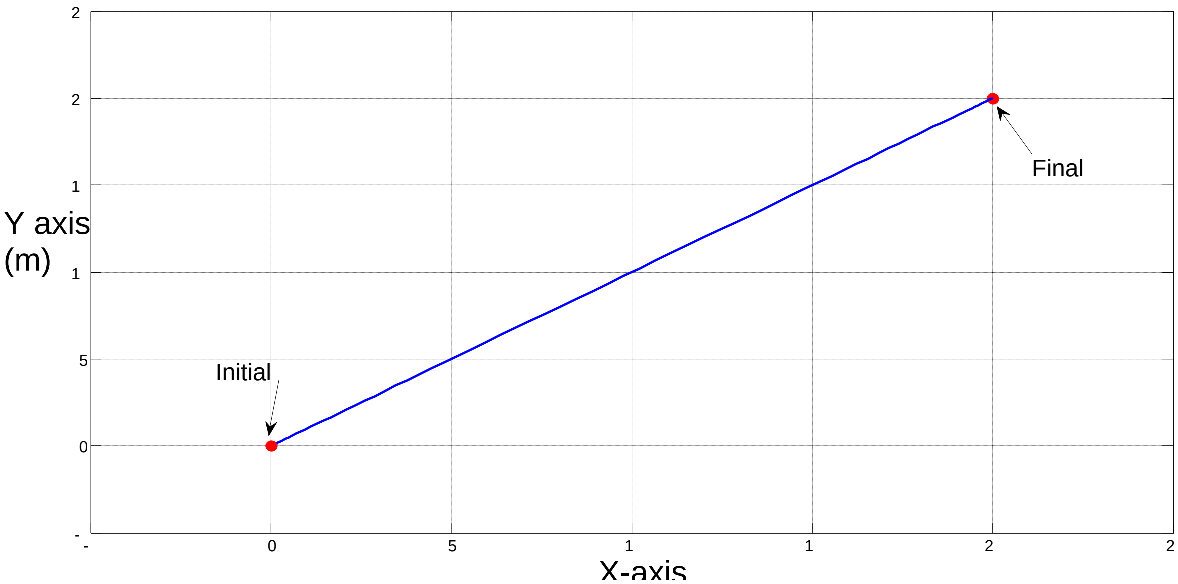

Figure 6. Basic First Order Polynomial

The straight line represented in the figure above, represents the satisfying of the initial boundary conditions.

While outlining the problem of an UNREP between the SCCC and a supply vessel, a straight line trajectory may not be the optimal trajectory to accomplish this objective, due to changes in course/speed, contact avoidance, and changes in rendezvous time. Based on these foreseen events, a curved path by be more appropriate in achieving this goal. As outlined by other research endeavors and as previously mentioned in this thesis, a curved path can be achieved by breaking the problem into smaller pieces or as is standard naval practice, put in additional waypoints to the final position or objective. While this seems simple enough, the added waypoints must then account for added boundary conditions of that particular leg, and an added time to the next leg must also be taken into account. While this is fine for typical naval applications, it does add to the overall computational time to achieve the optimal trajectory solution.

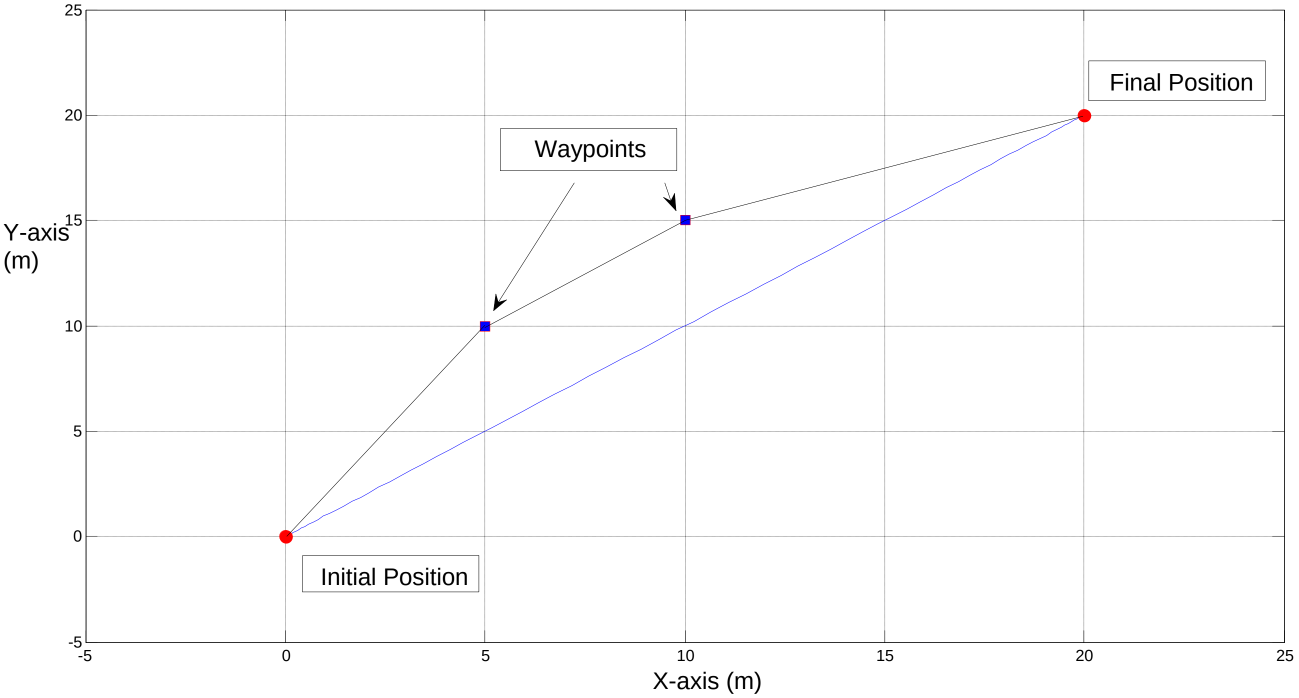

Figure 7. Curved Trajectory Using Waypoints

In Figure 7, two waypoints were added to achieve a curved trajectory between the initial and final positions. It is of interest to note that in order to approximate this new path, it must be represented as a polynomial. Additionally, an added waypoint is a direct corresponding result to an increase in the order of the initial polynomial. In this case the corresponding result would be a third order polynomial due to the addition of two waypoints. If this new curved path is represented in the form of a third order polynomial, it will produce a smooth curve which is very similar to the outline of the waypoints, while maintaining the existing boundary conditions. Using a direct method approach, the vessels controls would be taken from the produced trajectory and the required states would then be produce by using inverse dynamics.

Assuming the initial boundary conditions remain the same, this third order polynomial representation of the intended trajectory can be displayed as previously shown with only minor changes as shown below:

$$x(\tau) = P_X(\tau) = \sum_{i=0}^{3} a_i \tau^i \qquad (3\text{-}13)$$

$$y(\tau) = P_y(\tau) = \sum_{i=0}^{3} b_i \tau^i \qquad (3\text{-}14)$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & \tau_1 & \tau_1^2 & \tau_1^3 \\ 1 & \tau_1 & \tau_2^2 & \tau_2^3 \\ 1 & \tau_1 & \tau_3^2 & \tau_3^3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_f \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & \tau_1 & \tau_1^2 & \tau_1^3 \\ 1 & \tau_1 & \tau_2^2 & \tau_2^3 \\ 1 & \tau_1 & \tau_3^2 & \tau_3^3 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_f \end{bmatrix} \qquad (3\text{-}15)$$

By taking into account the first derivative of the first order polynomial equations, added path curvature flexibility will be increased allowing the user to utilize boundary conditions to gain the equations for the velocity.

An example of the process for a single high-order polynomial approximation for the coefficients of (A) in a “2+2” case are shown below.

$$x_i(\tau_0) = x_i(0) = x_{i0} \qquad x_i(\tau_f) = x_{if} \qquad (3\text{-}16)$$

$$x'_i(\tau_0) = x'_i(0) = x'_{i0} \qquad x'_i(\tau_f) = x'_{if} \qquad (3\text{-}17)$$

$$x''_i(\tau_0) = x''_i(0) = x''_{i0} \qquad x''_i(\tau_f) = x''_{if} \qquad (3\text{-}18)$$

$$x_i(\tau) = a_{i0} + a_{i1}\tau + a_{i2}\tau^2 + a_{i3}\tau^3 + a_{i4}\tau^4 + a_{i5}\tau^5 = \sum_{k=0}^{5} a_{ik}\tau^k \qquad \text{(3-19)}$$

$$x'_i(\tau) = a_{i1} + 2a_{i2}\tau + 3a_{i3}\tau^2 + 4a_{i4}\tau^3 + 5a_{i5}\tau^4 = \sum_{k=0}^{5} k a_{ik}\tau^{k-1} \qquad \text{(3-20)}$$

$$x''_i(\tau) = 2a_{i2} + 6a_{i3}\tau + 12a_{i4}\tau^2 + 20a_{i5}\tau^3 = \sum_{k=0}^{5} k(k-1) a_{ik}\tau^{k-2} \qquad \text{(3-21)}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 1 & \tau_f & \tau_f^2 & \tau_f^3 & \tau_f^4 & \tau_f^5 \\ 0 & 1 & 2\tau_f & 3\tau_f^2 & 4\tau_f^3 & 5\tau_f^4 \\ 0 & 0 & 2 & 6\tau_f & 12\tau_f^2 & 20\tau_f^3 \end{bmatrix} \begin{bmatrix} a_{i0} \\ a_{i1} \\ a_{i2} \\ a_{i3} \\ a_{i4} \\ a_{i5} \end{bmatrix} = \begin{bmatrix} x_{i0} \\ x'_{i0} \\ x''_{i0} \\ x_{if} \\ x'_{if} \\ x''_{if} \end{bmatrix} \qquad \text{(3-22)}$$

C. HYPOTHETICAL TRAJECTORIES

In order to create a random trajectory, the user must establish an initial and final position, which will be required to be stationary points. Using Matlab’s Symbolic Toolbox, one can determine the coefficients of the example from the previous section as shown below.

$$\begin{bmatrix} a_{i0} \\ a_{i1} \\ a_{i2} \end{bmatrix} = \begin{bmatrix} x_{i0} \\ x_{i1} \\ x_{i2} \end{bmatrix} \quad \begin{bmatrix} a_{i3} \\ a_{i4} \\ a_{i5} \end{bmatrix} = \begin{bmatrix} 3x''_{if} - 9x''_{i0} & -24x'_{if} - 36x'_{i0} & 60(x_{if} - x_{i0}) & 0 & 0 \\ 0 & 18x''_{if} - 12x''_{i0} & 84x'_{if} + 96x'_{i0} & -180(x_{if} - x_{i0}) & 0 \\ 0 & 0 & 10(x''_{if} - x''_{i0}) & -60(x'_{if} + x'_{i0}) & 120(x_{if} - x_{i0}) \end{bmatrix} \begin{bmatrix} \tau_f^{-1} \\ \tau_f^{-2} \\ \tau_f^{-3} \\ \tau_f^{-4} \\ \tau_f^{-5} \end{bmatrix} \qquad \text{(3-23)}$$

In order to conduct a single high-order polynomial approximation using a visual check, the inputs are set as below in the following table.

$x_{10} = 0$	$x_{20} = 0$	$x_{1f} = 1$	$x_{2f} = 1$
$x'_{10} = 0.2$	$x'_{20} = 1$	$x'_{1f} = 0.1$	$x'_{2f} = -1$
$x''_{10} = \text{var}$	$x''_{20} = 0.1$	$x''_{1f} = 0.1$	$x''_{2f} = 0.1$

Table 2. Example Inputs

where,

$$\tau_f = 1,,2,....,10$$
$$x''_{10} = \{-0.4;-0.1;0.2;0.5\}$$

The resulting plots are the visual confirmation check in meters.

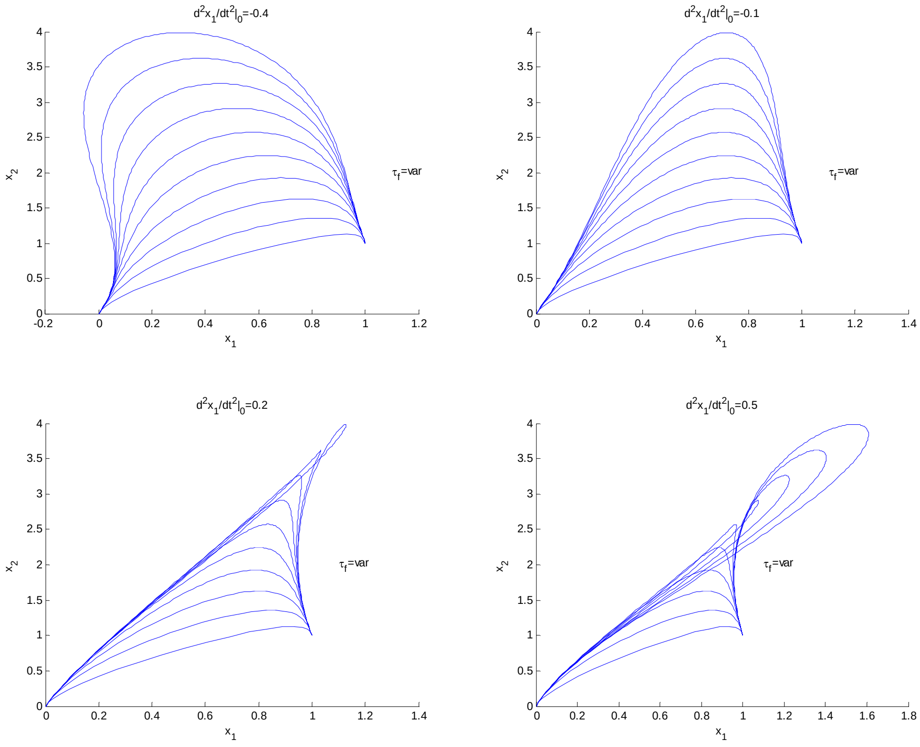

Figure 8. Visual Check

Utilizing the before mentioned techniques above, a random trajectory was established using the SCCC concepts of operation to establish the conditions for this trajectory. This random trajectory is shown below.

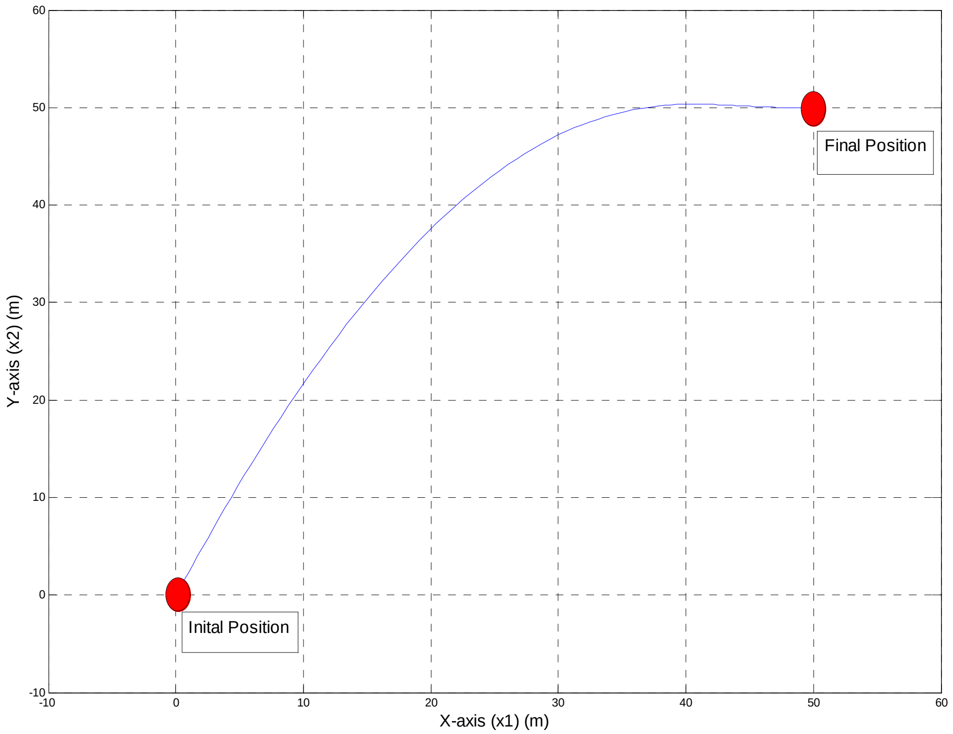

Figure 9. Random Trajectory Generation

IV. TRAJECTORY DEVELOPMENT AND VALIDATION

A. INDENTIFIYING CONTROL PARAMETERS & CONSTRAINTS

The SCCC UNREP with a supply vessel is a problem centered on navigation with the primary controlled parameters of concern being course and speed, which essentially makes this a two dimensional problem. For the propose of this thesis, the two vessels in question will be in open ocean not constrained by draft, however, the body of water may have above water obstructions such as oil platforms.

Viewing this problem, the constraints which come to mind are the rudder/speed rule of 15/15 for a total of 30 overall. Meaning, one could use a speed of 20 knots and a rudder of 10 knots for a total of 30. An additional constraint will deal with limiting speed and/or rudder commands when within approximately 350 feet of the supply vessel as to limit the possibility of collision.

B. INVERSE DYNAMICS

Inverse dynamics computes the states of the instantaneous position of the vessel along the virtual arc of a give trajectory. This process allows for reverse engineering of an executed trajectory in order to determine the required commands to achieve this maneuver while maintaining the necessary boundaries of constraints.

In order to achieve the fore mentioned process, it is of note that the parameters of the reference trajectory of a numerical solution are calculated in N points evenly distributed throughout the virtual arc, such that

$$\Delta\tau = \tau_f (N-1)^{-1} \qquad (4\text{-}1)$$

Time interval between two points is calculated as in equation (4-6)

$$\Delta t_{j-1} = t_j - t_{j-1} = 2\frac{\sqrt{(x_j - x_{j-1})^2 + (y_j - y_{j-1})^2}}{V_j + V_{j-1}} \qquad (4\text{-}2)$$

In order to transition from a measurement of position along the virtual arc to that of velocity, the kinematics for a navigational solution must be achieved and are shown below as an example.

Given:

$x_1(t), x_2(t)$ and therefore $\dot{x}_1(t), \dot{x}_2(t)$ and $\ddot{x}_1(t), \ddot{x}_2(t)$

Kinematic equations:

$$\dot{x}_1 = \frac{dx_1}{dt} = V\cos\Psi \qquad (4\text{-}3)$$

$$\dot{x}_2 = \frac{dx_2}{dt} = V\sin\Psi \qquad (4\text{-}4)$$

$$V = \sqrt{\dot{x}_1^2 + \dot{x}_2^2} \qquad (4\text{-}5)$$

$$\Psi = \arctan\frac{\dot{x}_2}{\dot{x}_1} \qquad (4\text{-}6)$$

Convert those to the virtual domain (where the reference trajectories are developed) using the virtual speed $\lambda = \frac{d\tau}{dt}$

$$x_1' = \frac{dx_1}{d\tau} = \frac{dx_1}{dt}\frac{dt}{d\tau} = \frac{1}{\lambda}V\cos\Psi \qquad (4\text{-}7)$$

$$x_2' = \frac{dx_2}{d\tau} = \frac{dx_2}{dt}\frac{dt}{d\tau} = \frac{1}{\lambda} V \sin\Psi \qquad \text{(4-8)}$$

We will set a separate 5th-order polynomial for $\lambda(\tau)$ similar to that of equation (3-29)

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & \tau_f & \frac{1}{2}\tau_f^2 & \frac{1}{6}\tau_f^3 & \frac{1}{12}\tau_f^4 & \frac{1}{20}\tau_f^5 \\ 0 & 1 & \tau_f & \frac{1}{2}\tau_f^2 & \frac{1}{3}\tau_f^3 & \frac{1}{4}\tau_f^4 \\ 0 & 0 & 1 & \tau_f & \tau_f^2 & \tau_f^3 \end{bmatrix} \begin{bmatrix} a_0^\lambda \\ a_1^\lambda \\ a_2^\lambda \\ a_3^\lambda \\ a_4^\lambda \\ a_5^\lambda \end{bmatrix} = \begin{bmatrix} \lambda_0 \\ \lambda_0' \\ \lambda_0'' \\ \lambda_f \\ \lambda_f' \\ \lambda_f'' \end{bmatrix} \qquad \text{(4-9)}$$

where λ_0'' and λ_f'' being varied parameters and λ_0, λ_0', λ_f and λ_f' defined as

$$\lambda_0 = V_0,\ \ \lambda_0' = 0,\ \ \lambda_f = V_f \ \text{ and } \ \lambda_f' = 0 \qquad \text{(4-10)}$$

Next, to address the constraints imposed on the control parameters (or in other words to account for the controllers dynamics) we also need to evaluate the following derivatives

$$\dot{V} = V'\lambda = \lambda'\sqrt{x_1'^2 + x_2'^2} + \frac{x_1'x_1'' + x_2'x_2''}{\sqrt{x_1'^2 + x_2'^2}}\lambda^2 \qquad \text{(4-11)}$$

$$\dot{\Psi} = \Psi'\lambda = \lambda\frac{x_1'x_2'' - x_1''x_2'}{x_1'^2}\cos^2\Psi \qquad \text{(4-12)}$$

C. OPTIMIZATION

In order to capitalize on the ability to optimize this problem a general block scheme approach is taken in order to accomplish this goal.

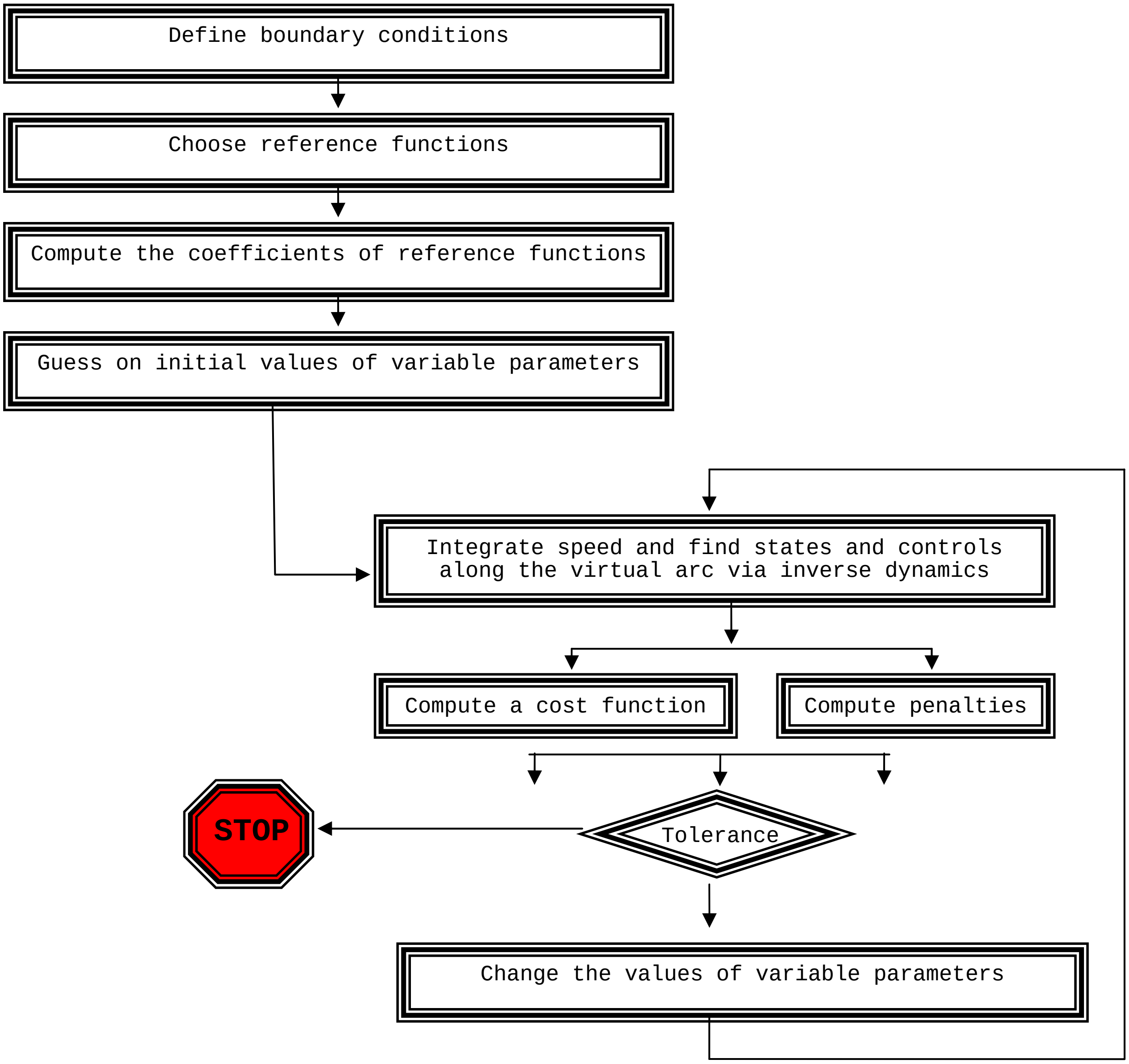
Define boundary conditions
Choose reference functions
Compute the coefficients of reference functions
Guess on initial values of variable parameters
Integrate speed and find states and controls along the virtual arc via inverse dynamics
Compute a cost function
Compute penalties
STOP
Tolerance
Change the values of variable parameters

Figure 10. General Block-Scheme

In short, Figure 10 helps to explain that this is a cycle and the values of each parameter are weighted to ensure the errors from the minimization function are continuously calculated until they are approximately zero, which will allow the UNREP rendezvous to be accomplished.

$$Perforamance\ Index = \sum Cost\ Functions + \sum Weight\ x\ Penalty \qquad (4\text{-}13)$$

Therefore the computational procedure, shown on Figure 10 looks as follows. We start from guessing on varied parameters, which are x''_{10}, x''_{20}, x''_{1f}, x''_{2f}, λ''_f, λ''_0 and τ_f. Then using the given initial and final conditions, which are shown in Table 4 below,

Initial Position x_1	Initial Position x_2	Final Position x_1	Final Position x_1
0	0	5000 yds	5000 yds

Table 3. Initial & Final Positions

one can compute the coefficients of the reference functions, which are based off of algebraic polynomials of degree “n” with the virtual arc “τ” as the argument. This allows for independent optimization of the velocity history along the trajectory. Next the coefficients of the reference functions are determined using MATLAB. The user then makes an initial guess on the values of the variable parameters.

Using inverse dynamics, the states and controls along the virtual arc are determined. From this point the cost function and penalties are computed and if these items are found to within tolerance then the computation stops. If the tolerances are not met, the values of the variable parameters are changed and the states and controls along the virtual arc are recalculated and the cycle repeats as shown in Figure 10.

Finally we estimate the performance index and evaluate penalties, the penalty weights are shown in Table 4.

Penalty Parameter	Weights
Time	10-3
Sway Velocity	10
Speed	10

Table 4. Penalty Weights

Additionally, the penalties are calculated by the equations shown below.

$$\text{Time Penalty} = (\text{total time} - T)^2 \qquad (4\text{-}14)$$

$$\text{Speed Penalty} = \max([0, \text{speed} - \text{speed}_{\max}])^2 \qquad (4\text{-}15)$$

V. RESULTS AND CONCLUSIONS

A. RESULTS

Using the technique described throughout this thesis, the following results were achieved for three Specialized Command and Control Crafts (SCCC) conducting an Underway Replenishment at Sea with a standard supply vessel.

For the three SCCCs, the initial and final velocities are the same, additionally for two out of three of the SCCCs, the initial angles are the same, however all three have the same final angles. The change in one of the initial angles was done to allow for a better visual image of the scenario. For these results the times of arrival were all the same, however to accommodate the supply vessel all SCCCs would have varying times of arrival in order for the supply ship to connect one vessel then move to the next one and so on. Additionally, once along side, the ability for the real time updating at a rate of (1Hz) would allow the supply vessel to alter course while all SCCCs maintain their stations.

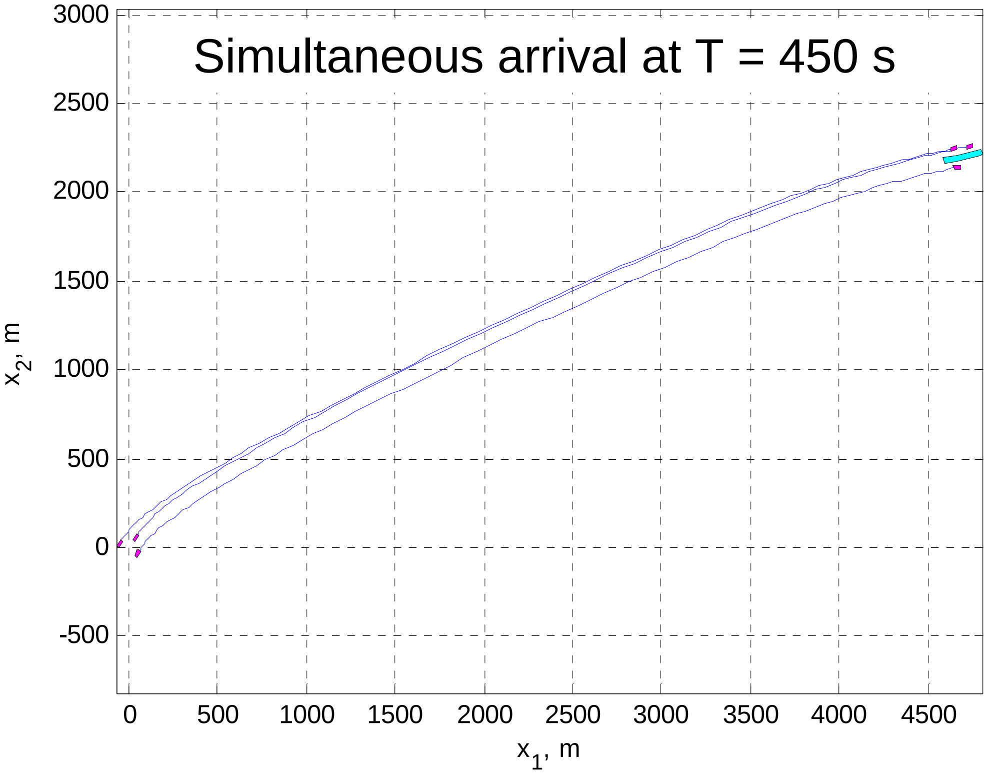

Figure 11. Multiple SCCC Scenario

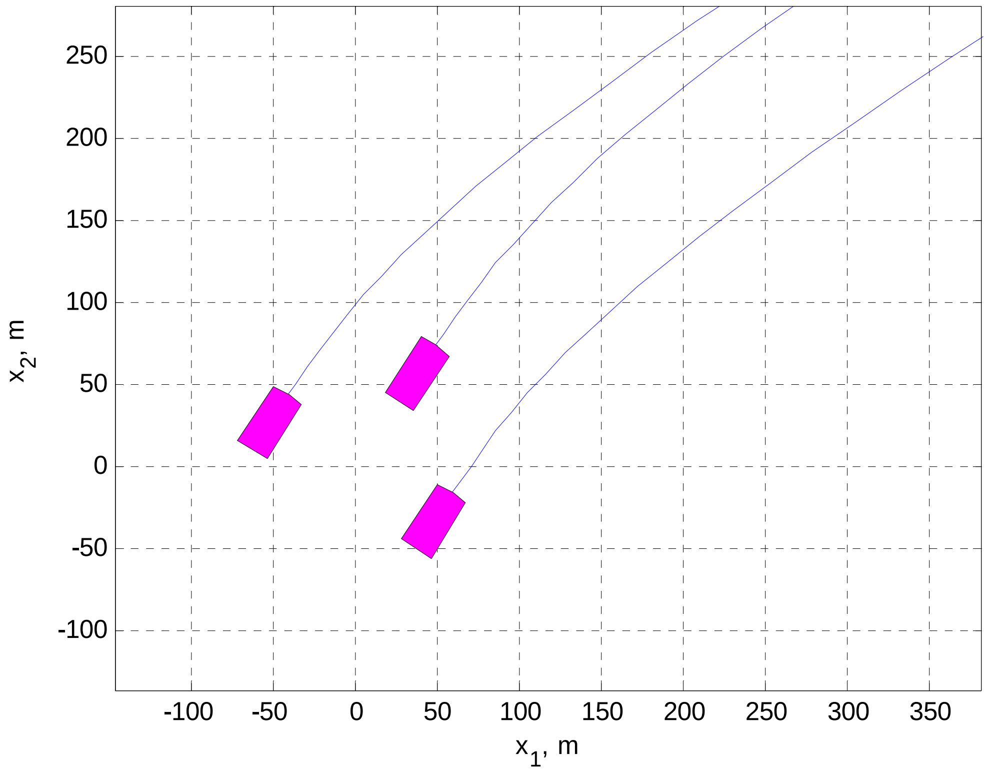

Figure 12.   Close-up of Initial Position of Multiple Scenario

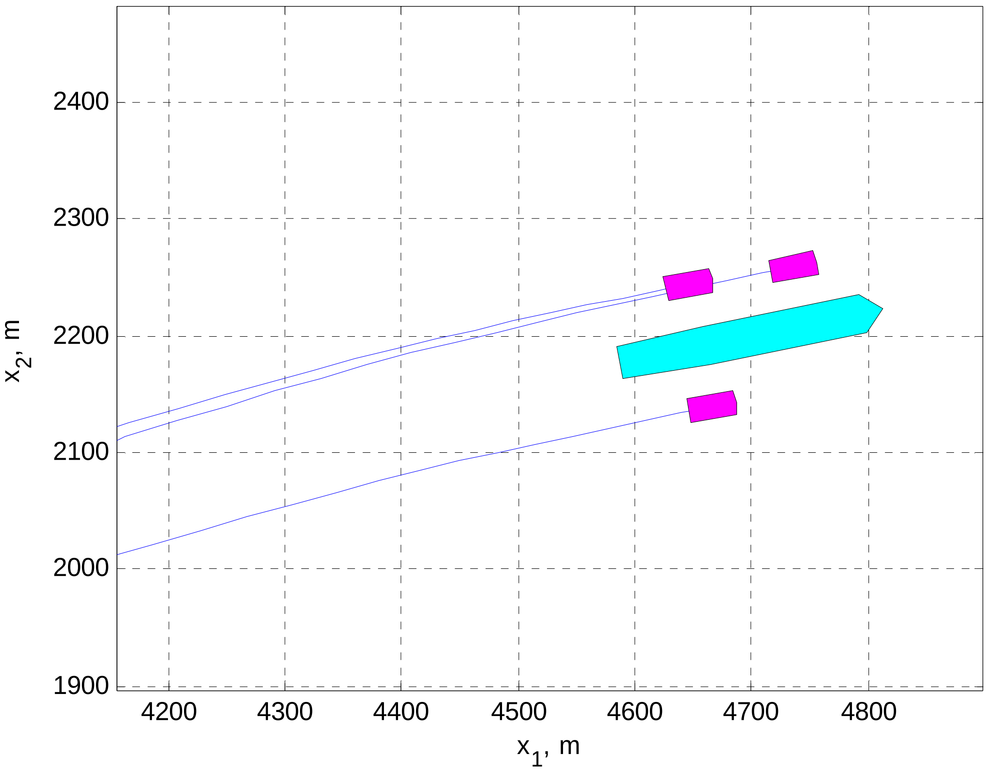

Figure 13. Close-up of Final Position of Multiple SCCC Scenario

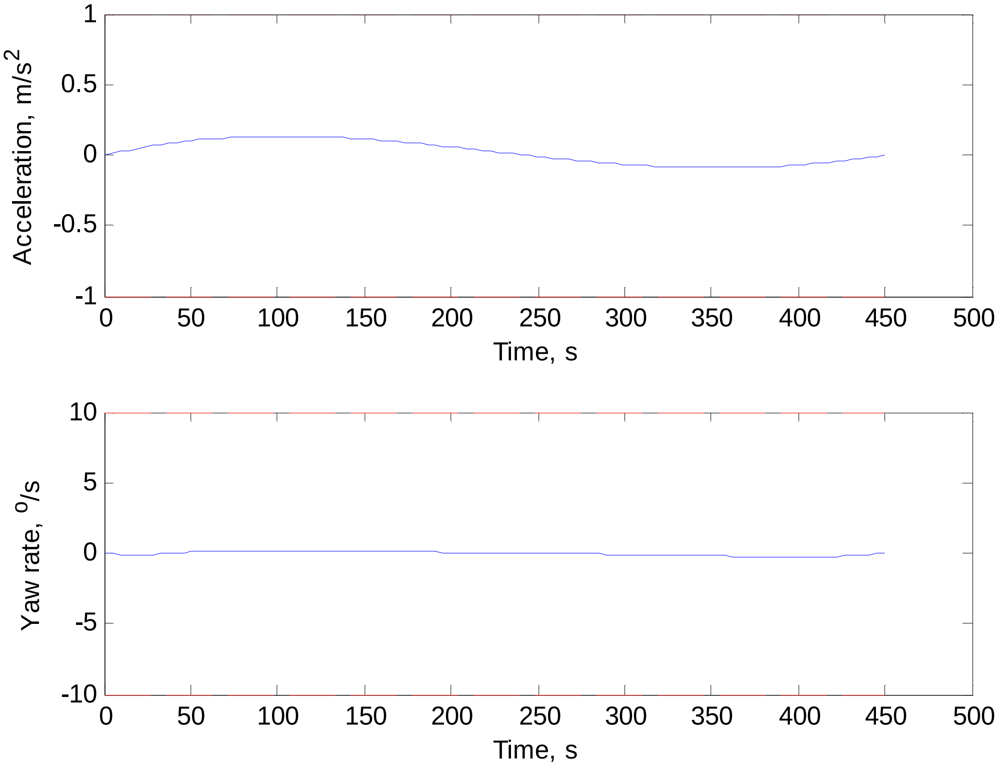

Figure 14. Controls for First SCCC

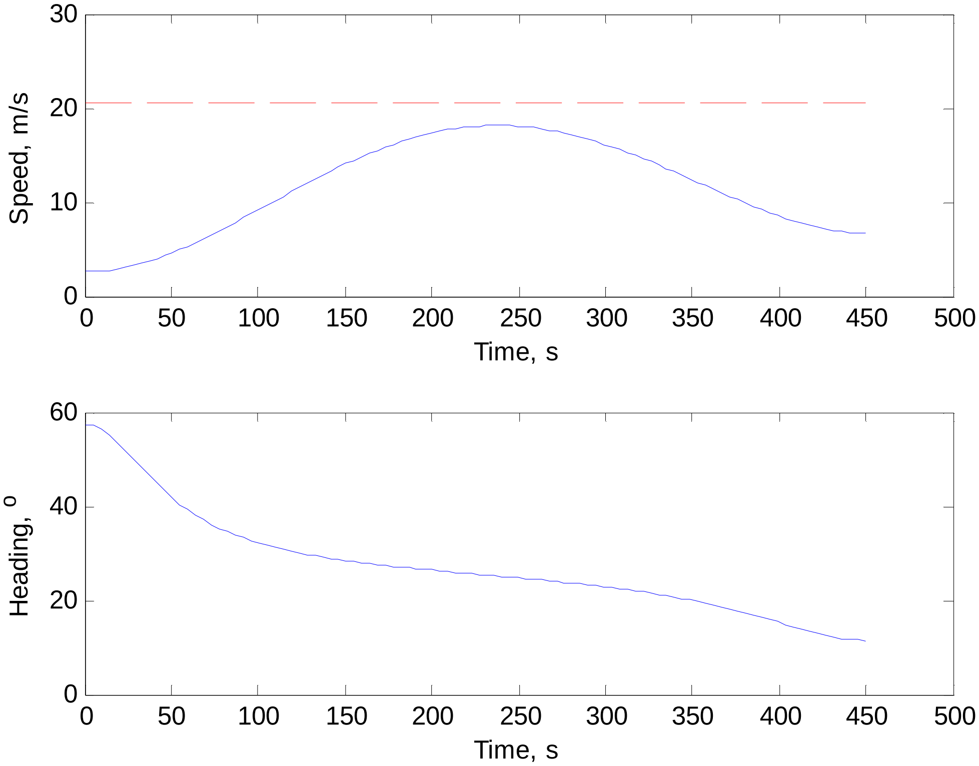

Figure 15. States for First SCCC

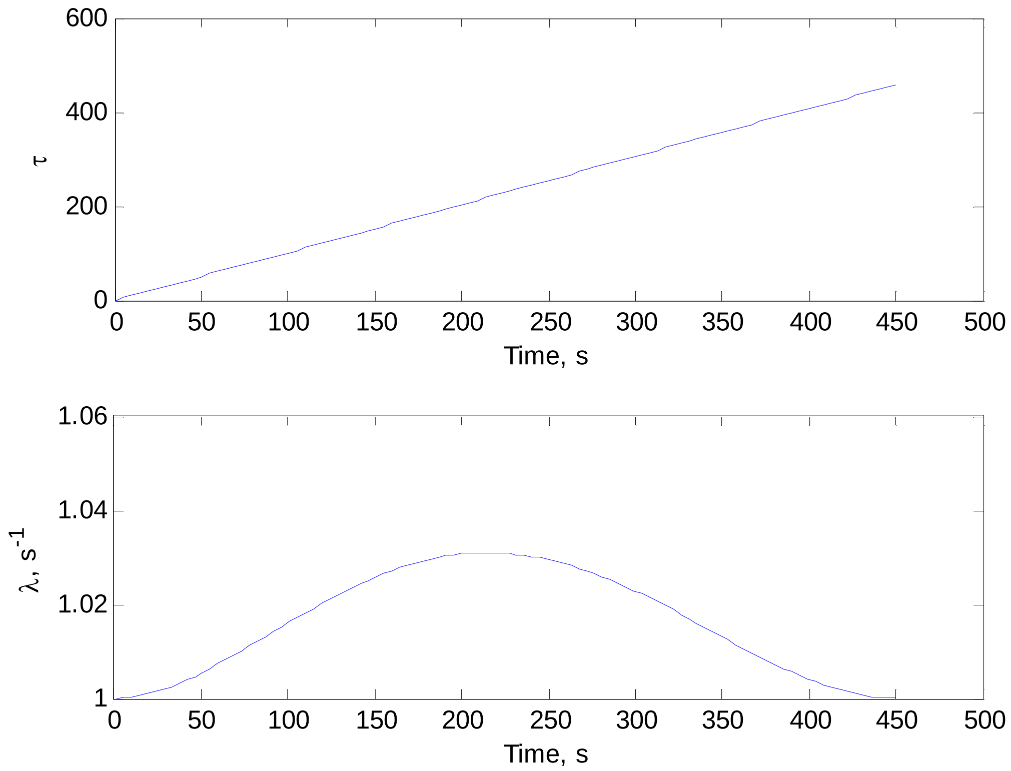

Figure 16. Virtual Domain Parameters for First SCCC

B. CONCLUSIONS

This model using the fore mentioned design method allows the problem boundary conditions to be satisfied beforehand; eliminates wild trajectories which decreases required computer computing times; allows for real time updating (1Hz) of the required outputs due to the speed at which calculations can be done; and allows for the implementation of multiple agents for collision-free motions.

Additionally benefits of this technology would be that it allows for the incorporation of a heads up display which could use standard maneuvers to build a database of near-optimal trajectories calculated beforehand. These near-optimal trajectories could all for the officer of the deck on board Naval vessels to not just mentally visualize the required commands, but also allow them to actually see a simulation of where them will end up, thus adding to the overall situational awareness. In principle, if this onboard computer was capable of doing the required updates often enough, the need for a traditional feedback controller would be unnecessary. The computer would then be capable of continuously regenerating from the vessels actual position instead of fighting the disturbance of the trajectories.

APPENDIX. MATLAB CODE

A. STARTMESCCC.M

```
% This is a main script
clear all, close all, clc
syms tf x0 xp0 xpp0 xf xpf xppf

%% Setting the boudary conditions
global posxi vxi posxf vxf
global posyi vyi posyf vyf
vi = 5*0.5144; vf = 13*0.5144;
posxi = -50;       posyi = 60;                   % initial position, m
posxf = 4650;     posyf = 2250;                  % final position, m
anglei = 0; anglef=0;
vxi   = vi*cos(anglei); vyi = vi*sin(anglei);    % components of initial
velocity
vxf   = vf*cos(anglef); vyf = vf*sin(anglef);    % components of final
velocity

%% Defining optimization problem
global Cost_T Fine_Speed Fine_Accel Fine_YawRate Psi_dot_max v_max a_max
T
global wv wvd wPsid
R=14;                     % lateral seperation distance, m
v_max=40*0.5144;          % maximum speed, m/s/s
a_max=10.7;               % maximum acceleration, m/s/s
%T  = 2*sqrt((posxf-posxi)^2+(posyf-posyi)^2)/(vi+vf)/2;  %
predetermined time of arrival, s
T = 550;
Psi_dot_max = 10*pi/180;  % maximum yaw rate, rad/s
wv  = 10 ;                % weighting coefficient for speed
wvd  = 10;                % weighting coefficient for accelaration
wPsid = 10;               % weighting coefficient for yaw rate

%% Guessing on the varied parameters
guess(1)=0.02*sqrt((posxf-posxi)^2+(posyf-posyi)^2); % virtual arc
length
guess(2)=rand*0.01;        % initial acceleration in x
guess(3)=rand*-0.0001;     % initial acceleration in y
guess(4)=rand*-0.0001;     % final acceleration in x
guess(5)=rand*0.001;       % final acceleration in y
guess(6)=rand*-0.0001;     % initial acceleration in lambda
guess(7)=rand*0.0001;      % final acceleration in lambda

%% Defining coefficients (units are commented)
%DefineSCCC
%% Compute Coefficients
global a M
A=[1 0  0      0      0       0;
   0 1  0      0      0       0;
   0 0  1      0      0       0;
   1 tf tf^2/2 tf^3/6 tf^4/12 tf^5/20;
   0 1  tf     tf^2/2 tf^3/3  tf^4/4;
   0 0  1      tf     tf^2  tf^3];
b=[x0 xp0 xpp0 xf xpf xppf]';
a=A\b;
a=collect(a,tf);
M=length(a);

%% Calling the optimization routine
opt=optimset('Display','iter','TolX',1e-4,'TolFun',1e-4,'MaxIter',10);
[guess_opt,fval,exitflag]=fminsearch('Trajectory',guess,opt);
%Trajectory(guess);

%% Displaying cost function and penalties
fprintf('Cost function                 :  %6.2g\n',Cost_T)
fprintf(' Penalty in speed           :  %6.2g\n',Fine_Speed)
fprintf(' Penalty in acceleration    :  %6.2g\n',Fine_Accel)
fprintf(' Penalty in yaw rate        :  %6.2g\n\n',Fine_YawRate)

%% Displaying optimal parameters
%guess_opt=guess;
fprintf('Arc lenght = %6.2f\n',guess_opt(1))
fprintf('             initial accel    final accel\n')
fprintf('along x  :    %6.2e        %6.2e\n',[guess_opt(2)
guess_opt(3)])
fprintf('along y  :    %6.2e        %6.2e\n',[guess_opt(4)
guess_opt(5)])
fprintf('in lambda:    %6.2e        %6.2e\n',[guess_opt(6)
guess_opt(7)])

%% Plotting the results
PlotResults
```

B. TRAJECTORY.M

```
function PI=trajectory(guess)
%% This function computes states and controls for the current guess
global a M
global tf x0 xp0 xpp0 xf xpf xppf
syms tf x0 xp0 xpp0 xf xpf xppf
global posxi vxi posxf vxf
global posyi vyi posyf vyf
global t x1 x2 v Psi tau lam v_dot Psi_dot

%% Current values of varied parameters
tauf =guess(1);   % virtual arc length
accxi=guess(2);   % initial acceleration in x
accyi=guess(3);   % initial acceleration in y
accxf=guess(4);   % final acceleration in x
accyf=guess(5);   % final acceleration in y
accli=guess(6);   % initial acceleration in lambda
acclf=guess(7);   % final acceleration in lambda
```

```
%% Defining coordinates in N nodes in the virtual domain
a1 = subs(a,{'x0','xp0','xpp0','xf','xpf','xppf','tf'},...

{posxi,vxi,accxi,posxf,vxf,accxf,tauf});
a2 = subs(a,{'x0','xp0','xpp0','xf','xpf','xppf','tf'},...

{posyi,vyi,accyi,posyf,vyf,accyf,tauf});
a3 = subs(a,{'x0','xp0','xpp0','xf','xpf','xppf','tf'},...

{1,0,accli,1,0,acclf,tauf});
ax1=diag([1,1,1/2,1/6,1/12,1/20])*a1;
ax2=diag([1,1,1/2,1/6,1/12,1/20])*a2;
ax3=diag([1,1,1/2,1/6,1/12,1/20])*a3;

 tau=linspace(0,tauf);

 for i=1:M
   cx1(i)=ax1(M+1-i);
   cx2(i)=ax2(M+1-i);
   cx3(i)=ax3(M+1-i);
 end
x1  = polyval(cx1,tau);
x2  = polyval(cx2,tau);
lam = polyval(cx3,tau);

%% Defining coordinates' derivatives in N nodes in the virtual domain
    cx1_prime = cx1.*[5:-1:0]*eye(6,5);
    cx2_prime = cx2.*[5:-1:0]*eye(6,5);
    cx3_prime = cx3.*[5:-1:0]*eye(6,5);
    x1_prime = polyval(cx1_prime,tau);
    x2_prime = polyval(cx2_prime,tau);
   lam_prime = polyval(cx3_prime,tau);

%% Defining coordinates' second-order derivatives in N nodes
    ex1_2prime=cx1_prime.*[4:-1:0]*eye(5,4);
    ex2_2prime=cx1_prime.*[4:-1:0]*eye(5,4);
    x1_2prime=polyval(ex1_2prime,tau);
    x2_2prime=polyval(ex2_2prime,tau);

%% Computing the states and controls using Inverse Dynamics
N=length(x1);
    del_tau = tauf/(N-1);
    t(1)    = 0;                            % time
    v(1)    = sqrt(vxi^2+vyi^2);            % initial speed, m/s
    Psi     = atan2(x2_prime,x1_prime);     % heading, rad
for j=2:N
sq         = sqrt((x1_prime(j))^2+(x2_prime(j))^2);
v(j)       = lam(j)*sq;                     % speed, m/s
dt         = 2*sqrt((x1(j)-x1(j-1))^2+(x2(j)-x2(j-1))^2)/(v(j)+v(j-1));
t(j)       = t(j-1)+dt;
v_dot(j)   = lam_prime(j)*sq+...

lam(j)^2*((x1_prime(j)*x1_2prime(j))+(x2_prime(j)*x1_2prime(j)))/sq;
Psi(j)     = atan2(x2_prime(j),x1_prime(j));
```

```
Psi_dot(j) = (x1_prime(j)*x2_2prime(j)-x1_2prime(j)*x2_prime(j))...
             /x1_prime(j)^2*cos(Psi(j))^2;
end
PI = PerformanceIndex;
return

function PI=PerformanceIndex
%% This function computes the combined performance index
global t x1 x2 v Psi tau lam v_dot Psi_dot
global Cost_T Fine_Speed Fine_Accel Fine_YawRate Psi_dot_max v_max a_max
T
global wv wvd wPsid
    Cost_T          = (t(end)-T)^2;
    Fine_Speed      = max([0,abs(v)-v_max])^2;
    Fine_Accel      = max([0,(abs(v_dot)-a_max)])^2;
    Fine_YawRate    = max([0,(abs(Psi_dot)-Psi_dot_max)])^2;
%    Fine_A          = max(0,R-min(sqrt((x1-posxfr).^2+(x2-
posyfr).^2)))^2;
PI = Cost_T + wv*Fine_Speed + wvd*Fine_Accel + wPsid*Fine_YawRate;
return
```

C. PLOTRESULTS.M

```
%% This script plots the results of optimization
global t x1 x2 v Psi tau lam v_dot Psi_dot
global Cost_T Fine_Speed Fine_Accel Fine_YawRate Psi_dot_max v_max a_max
T

%% Bird-eye view
figure('Name','2D View')
title('Optimal Trajectory')
plot(x1,x2)
xlabel('x_1, m'), ylabel('x_2, m')
axis equal, grid on, hold on
plot(x1(1),x2(1),'rO')
plot(x1(end),x2(end),'rO')

%% Plotting time histories
figure('Name','States')
subplot(2,1,1)
plot(t,v), hold
xlabel('Time, s'), ylabel('Speed, m/s')
plot([0 t(end)],v_max*[1 1],'r--')
subplot(2,1,2)
plot(t,Psi*180/pi)
xlabel('Time, s'), ylabel('Heading, ^o')

figure('Name','Controls')
subplot(2,1,1)
plot(t,v_dot), hold
xlabel('Time, s'), ylabel('Acceleration, m/s^2')
plot([0 t(end)],a_max*[1 1],'r--')
plot([0 t(end)],-a_max*[1 1],'r--')
subplot(2,1,2)
```

```
plot(t,Psi_dot*180/pi), hold
xlabel('Time, s'), ylabel('Yaw rate, ^o/s')
plot([0 t(end)],Psi_dot_max*[1 1]*180/pi,'r--')
plot([0 t(end)],-Psi_dot_max*[1 1]*180/pi,'r--')

figure('Name','Virtual Domain Parameters')
subplot(2,1,1)
plot(t,tau)
xlabel('Time, s'), ylabel('\tau')
subplot(2,1,2)
plot(t,lam)
xlabel('Time, s'), ylabel('\lambda, s^{-1}')
```

D. TEST.M

```
close all
SupplyShip(4700,2200,0.2)
SCCCship(50,0,1)
SCCCship(40,90,1)
SCCCship(-50,60,1)
```

E. SUPPLYSHIP.M

```
function SupplyShip(X,Y,Psi)
% X,Y - the coordinates of the ship's center in the local tangent plane
in meters
% Psi - the orientation of the ship in the local tangent plane in
radians
SCCCx=[0 250 700 754.6 700 250 0]*0.3048;
SCCCy=[100 107 107 53.7 0 0 7]*0.3048;
  SCCCx=SCCCx-mean(SCCCx);
  SCCCy=SCCCy-mean(SCCCy);
  len=sqrt(SCCCx.^2+SCCCy.^2);
  ang=atan2(SCCCy,SCCCx);
    SCCCx=X+len.*cos(Psi-ang);
    SCCCy=Y+len.*sin(Psi-ang);
patch(SCCCx,SCCCy,'c')
axis equal
```

F. SCCCSHIP.M

```
function SCCCship(X,Y,Psi)
% X,Y - the coordinates of the ship's center in the local tangent plane
in meters
% Psi - the orientation of the ship in the local tangent plane in
radians
SCCCx=[0 130 135 130 0]*0.3048;
SCCCy=[68 68 34 0 0]*0.3048;
  SCCCx=SCCCx-mean(SCCCx);
  SCCCy=SCCCy-mean(SCCCy);
  len=sqrt(SCCCx.^2+SCCCy.^2);
  ang=atan2(SCCCy,SCCCx);
```

```
    SCCCx=X+len.*cos(Psi-ang);
    SCCCy=Y+len.*sin(Psi-ang);
patch(SCCCx,SCCCy,'m')
axis equal
```

LIST OF REFERENCES

[1] Office of the Chief of Naval Operations. *Riverine Force, Concept of Operations for the 21^{st} Century.* March 2006

[2] Review of Navy’s Riverine Capability; Resource Implications for Combat Support/Combat Service Support and TSC, Center for Naval Analyses

[3] “Lecture summary 14, Decision making and Tradeoff Studies”; class notes for TSSE 4001, Department of Mechanical Engineering, Naval Postgraduate School, Spring 2007”

[4] TSSE Group 2004, “High-Speed Assault Connector for Amphibious Seabasing Operations and Joint Expeditionary Logistics,” Technical Report, Naval Postgraduate School, Monterey, CA, USA, 2004

[5] SEA-11, *Riverine Sustainment 2012*, Naval Postgraduate School, June 2007

[6] TSSE Group 2007, “Tiberinus Class Specialized Command and Control Craft for the Twenty First Century Riverine Force,” Technical Report, Naval Postgraduate School, Monterey, CA, USA, 2007

[7] Healey, Anthony J., Dynamics of Marine Vehicles (ME-4823), Class Notes, Naval Postgraduate School, Winter 2001

[8] Ogata, Katsuhiko, Modern Control Engineering, fourth ed., Pearson Education, Inc, 2004

[9] Yakimenko, Oleg, Direct Methods for Rapid Prototyping of Near-Optimal Aircraft Trajectories, Vol. 23, No. 5, September-October 2000

[10] Yakimenko, Oleg, Direct Methods for Rapid Prototyping (ME-4902), Class Notes, October 2007

[11] Pratt, Douglas, “AUV Rendezvous Using Real-Time Near Optimal Trajectories for Submarine Recovery,” NPS Thesis, December 2007

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Mechanical Engineering Department Chair, Code ME
   Distinguished Prefessor Anthony J. Healey
   Naval Postgraduate School
   Monterey, California

4. Mechanical Engineering Curriculum, Code 34
   Naval Postgraduate School
   Monterey, California

5. Undersea Warfare Department Chair, Code USW
   Professor Donald Brutzman
   Naval Postgraduate School
   Monterey, California

6. Undersea Warfare Curriculum, Code 75
   Naval Postgraduate School
   Monterey, California

7. Professor Oleg Yakimenko
   Naval Postgraduate School
   Monterey, California

8. Professor Fotis Papoulias
   Naval Postgraduate School
   Monterey, California

9. LT William F. Addison II
   Surface Warfare Officer School
   Newport, Road Island